# Using Stochastic Programming Problem Structure to Gain Computational Efficiency

John R. Birge

Northwestern University

# Outline

- General Model – Observations
- Overview of approaches
- Factorization/sparsity (interior point/barrier)
- Decomposition
- Lagrangian methods
- Conclusions

Theme: taking advantage of repeated problem structure can yield significant computational savings.

# General Stochastic Programming Model: Discrete Time

- Find $x=(x_1,x_2,\ldots,x_T)$ and p to

**minimize** $E_p [ \Sigma_{t=1}^T f_t(x_t,x_{t+1},p) ]$
**s.t.** $x_t \in X_t$, $x_t$ nonanticipative p in P (distribution class)
$P[ h_t (x_t,x_{t+1},p_t) <= 0 ] >= a$ (chance constraint)

**General Approaches:**

- **Simplify distribution (e.g., sample) and form a mathematical program:**
- **Solve step-by-step (dynamic program)**
- **Solve as single large-scale optimization problem**
- **Use iterative procedure of sampling and optimization steps**

# Simplified Finite Sample Model

- Assume p is fixed and random variables represented by sample $\xi^i_t$ for t=1,2,..,T, i=1,…,$N_t$ with probabilities $p^i_t$ ,a(i) an *ancestor* of i, then model becomes (no chance constraints):

$$\textbf{minimize} \quad \sum_{t=1}^{T} \sum_{i=1}^{N_t} p^i_t \, f_t(x^{a(i)}_t, x^i_{t+1}, \xi^i_t)$$
$$\textbf{s.t.} \quad x^i_t \in X^i_t$$

Observations?

- Problems for different i are similar – solving one may help to solve others
- Problems may decompose across i and across t yielding
  - smaller problems (that may scale linearly in size)
  - opportunities for parallel computation.

# Outline

- General Model – Observations
- Overview of approaches
- Factorization/sparsity (interior point/barrier)
- Decomposition
- Lagrangian methods
- Conclusions
- 

# Solving As Large-scale Mathematical Program

- Principles:
  - discretization leads to mathematical program but large-scale
  - use standard methods but exploit structure
- Direct methods
  - take advantage of sparsity structure
    - some efficiencies
  - use similar subproblem structure
    - greater efficiency
- Size
  - unlimited (infinite numbers of variables)
  - still solvable (caution on claims)

# Standard Approaches

- Sparsity Structure Advantage
  - Partitioning
  - Basis Factorization
  - Interior Point Factorization
- Similar/Small Problem Advantage
  - DP Approaches: Decomposition
    - Benders, L-shaped (Van Slyke – Wets)
    - Dantzig-Wolfe (Primal Version)
    - Regularized (Ruszczynski)
    - Various Sampling Schemes (Higle/Sen Stochastic Decomposition, Abridge Nested Decomposition)
  - Lagrangian Methods

# Outline

- General Model – Observations
- Overview of approaches
- Factorization/sparsity (interior point/barrier)
- Decomposition
- Lagrangian methods
- Conclusions

# Sparsity Methods: Stochastic Linear Program Example

- Two-stage Linear Model:

  $X_1 = \{x_1 | A\, x_1 = b,\ x_1 >= 0\}$

  $f_0(x_0,x_1)=c\, x_1$

  $f_1\,(x_1,x_2^i,\xi_2^i) = q\, x_2^i$ if $T\, x_1 + W\, x_2^i = \xi_2^i,$

  $x_2^i >= 0;\ + \infty$ otherwise

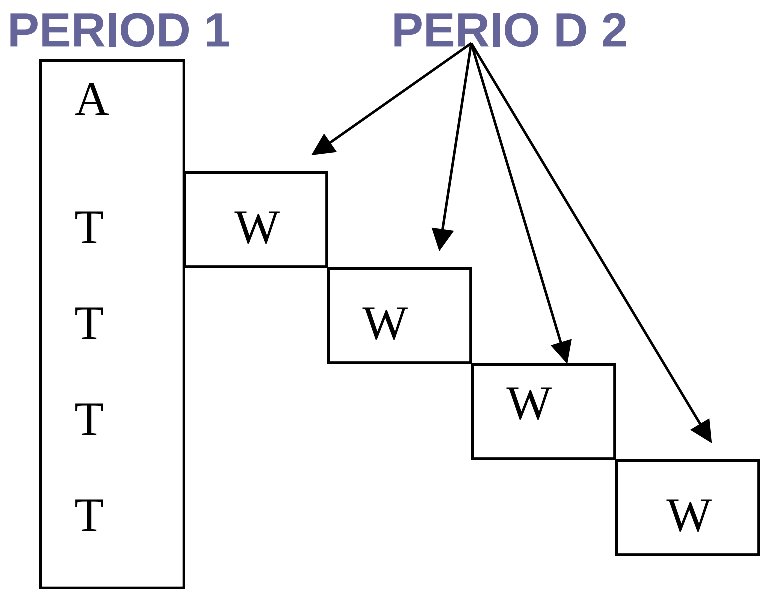
- Result: $\min c\, x_1 + \Sigma_{i=1}^{N1}\, \mathbf{p}_2^i\, q\, x_2^i$

  s. t. $A\, x_1 = b,\ x_1 >= 0$

  $T\, x_1 + W\, x_2^i = \xi_2^i,\ x_2^i >= 0$

# LP-Based Methods

- Using basis structure:




= A’

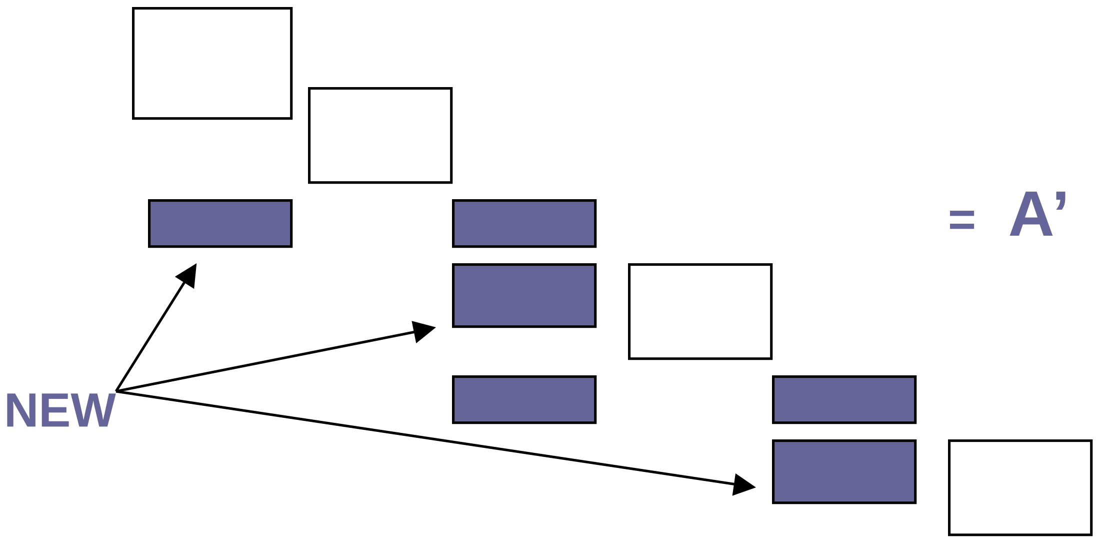
- Modest gains for simplex

**Interior Point Matrix Structure**

$A'D^2A'^T=$  COMPLETE FILL-IN

# Alternatives For Interior Points

- Variable splitting (Mulvey Et Al.)
  - Put in explicit *nonanticipativity constraints*

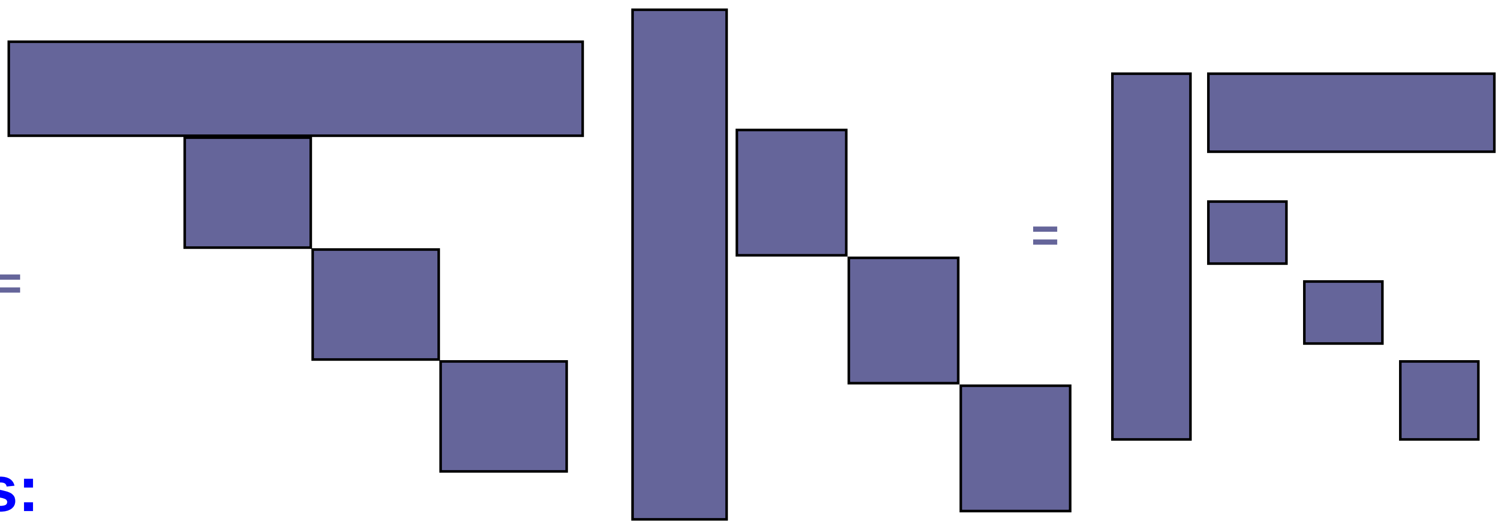



- **Result**
  - **Reduced fill-in but larger matrix**

# Other Interior Point Approaches

- Use of dual factorization or modified Schur complement

$A'^T D^2 A' =$




**Results:**

- **Speedups of 2 to 20**
- **Some instability => Indefinite system (Vanderbei et al. Czyzyk et al.)**

# Outline

- General Model – Observations
- Overview of approaches
- Factorization/sparsity (interior point/barrier)
- Decomposition
- Lagrangian methods
- Conclusions

Theme: taking advantage of repeated problem structure can yield significant computational savings.

# Similar/small Problem Structure: Dynamic Programming View

- Stages: $t=1,\ldots,T$
- States: $x_t \to B_t x_t$ (or other transformation)
- Value Function:

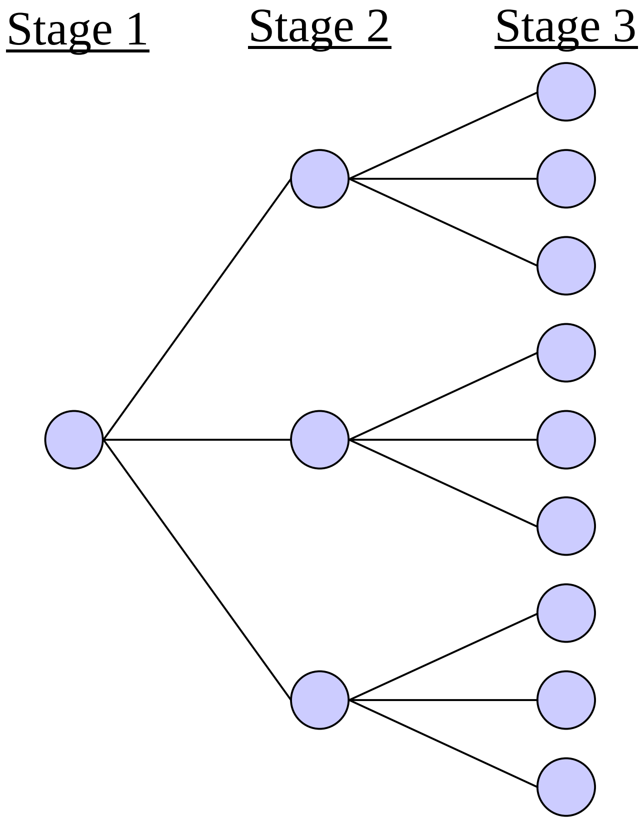
  $\mathcal{Q}_t(x_t) = E[Q_t(x_t,\xi_t)]$ where

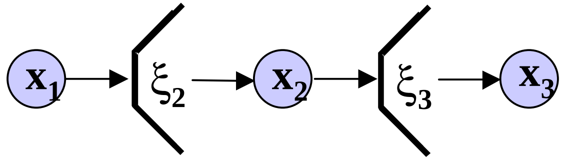
  $\xi_t$ is the random element and

  $Q_t(x_t,\xi_t) = \min f_t(x_t,x_{t+1},\xi_t) + \mathcal{Q}_{t+1}(x_{t+1})$

  s.t. $x_{t+1} \in X_{t+1t}(,\xi_t)$ $x_t$ given
- Solve : iterate from T to 1

# Linear Model Structure




$$\begin{array}{lll} \min & c_1 x_1 + Q_2(x_1) & \\ s.t. & W_1 x_1 = & h_1 \\ & x_1 \geq & 0 \end{array}$$

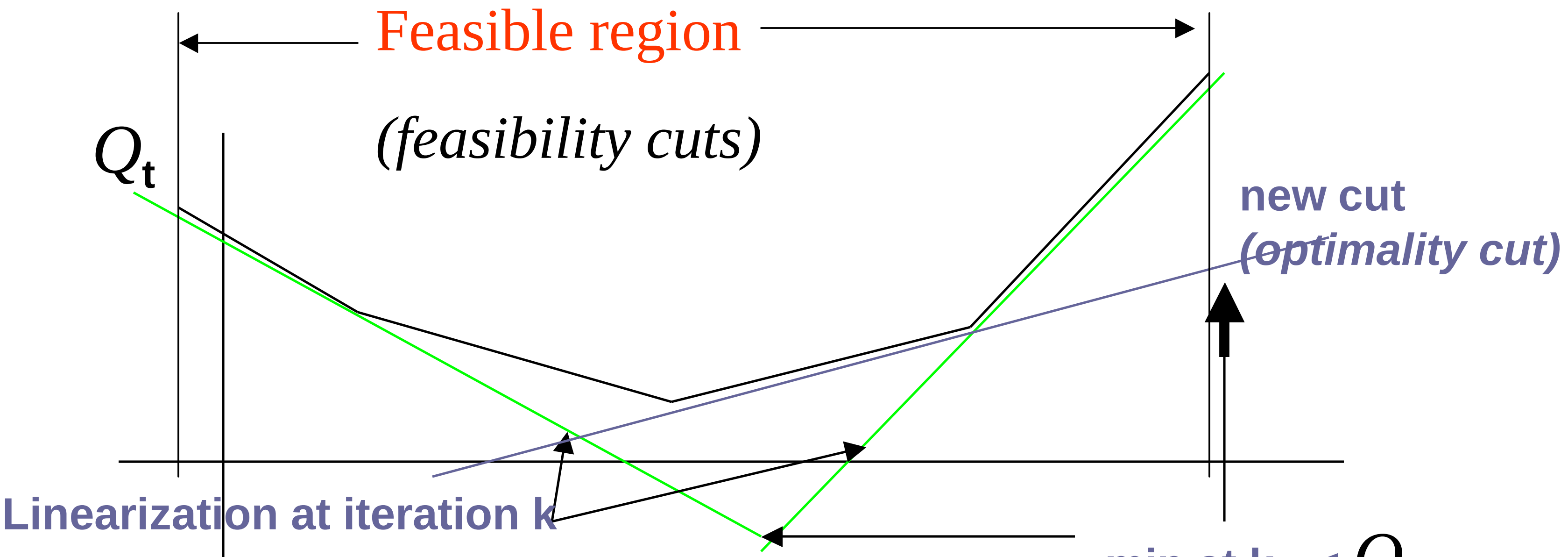
$$Q_t\left(x_{t-1,a(k)}\right) = \sum_{\xi_{t,k} \in \Xi_t} prob(\xi_{t,k}) \mathrm{Q}_{t,k}\left(x_{t-1,a(k)}, \xi_{t,k}\right)$$

$$\mathrm{Q}_{t,k}\left(x_{t-1,a(k)}, \xi_{t,k}\right) = \begin{array}{lll} \min & c_t(\xi_{t,k}) x_{t,k} + Q_{t+1}(x_{t,k}) & \\ s.t. & W_t x_{t,k} = & h_t(\xi_{t,k}) - T_{t-1}(\xi_{t,k}) x_{t-1,a(k)} \\ & x_{t,k} \geq & 0 \end{array}$$

- $Q_{N+1}(x_N) = 0$, for all $x_N$,
- $Q_{t,k}(x_{t-1,a(k)})$ is a piecewise linear, convex function of $x_{t-1,a(k)}$

# Decomposition Methods

- Benders Idea
  - Form an outer linearization of $Q_t$
  - Add *Cuts* On Function :

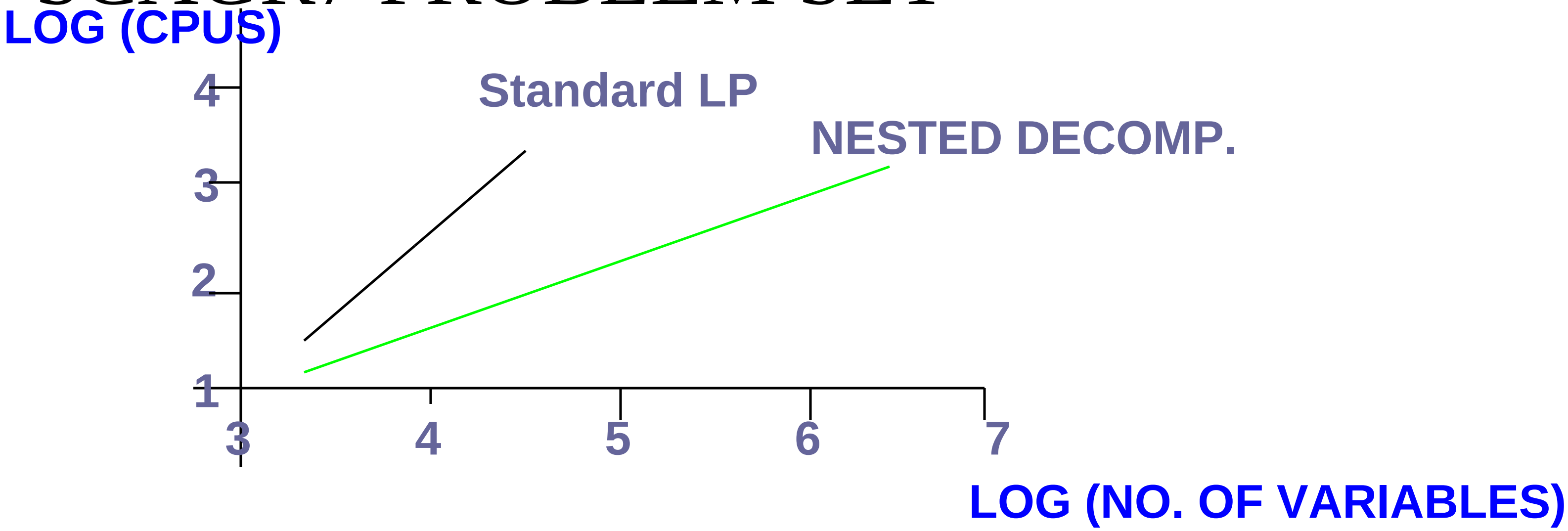



**Use at each stage to approximate value function**

- **Iterate between stages until all min =** $Q_t$

# Sample Results

- SCAGR7 PROBLEM SET




**PARALLEL:** **60-80% EFFICIENCY IN SPEEDUP**

**OTHER PROBLEMS: SIMILAR RESULTS**

# Decomposition Enhancements

- Optimal basis repetition
  - Take advantage of having solved one problem to solve others
  - Use *bunching* to solve multiple problems from root basis
  - *Share* bases across levels of the scenario tree
  - Use solution of single scenario as *hot start*
- Multicuts
  - Create cuts for each descendant scenario
- Regularization
  - Add quadratic term to keep close to previous solution
- Sampling
  - Stochastic decomposition (Higle/Sen)
  - Importance sampling (Infanger/Dantzig/Glynn)
  - Multistage (Pereira/Pinto, Abridged ND)

# Multistage: Pereira-Pinto Method

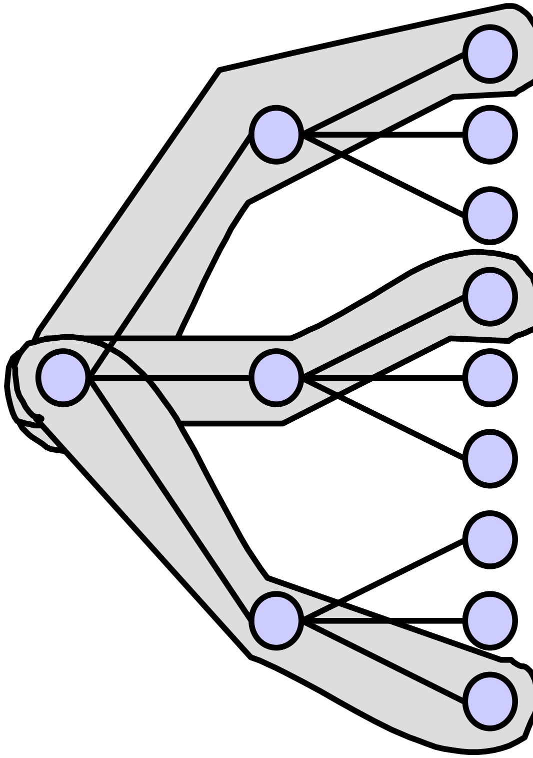
- Incorporates sampling into the general framework of the Nested Decomposition algorithm
- Assumptions:
  - relatively complete recourse
    - no feasibility cuts needed
  - serial independence
    - an optimality cut generated for any Stage t node is valid for all Stage t nodes
- Successfully applied to multistage stochastic water resource problems

# Pereira-Pinto Method

1. Randomly select *H N*-Stage scenarios
2. Starting at the root, a forward pass is made through the sampled portion of the scenario tree (solving ND subproblems)
3. A statistical estimate of the first stage objective value $\bar{z}$ is calculated using the total objective value obtained in each sampled scenario

the algorithm terminates if current first stage objective value $c_1 x_1 + \theta_1$ is within a specified confidence interval of $\bar{z}$

4. Starting in sampled node of Stage t = N-1, solve all Stage t+1 descendant nodes and construct new optimality cut. Repeat for all sampled nodes in Stage t, then repeat for t = t - 1



# Pereira-Pinto Method

- Advantages
  - significantly reduces computation by eliminating a large portion of the scenario tree in the forward pass
- Disadvantages
  - requires a complete backward pass on all sampled scenarios
    - not well designed for bushier scenario trees

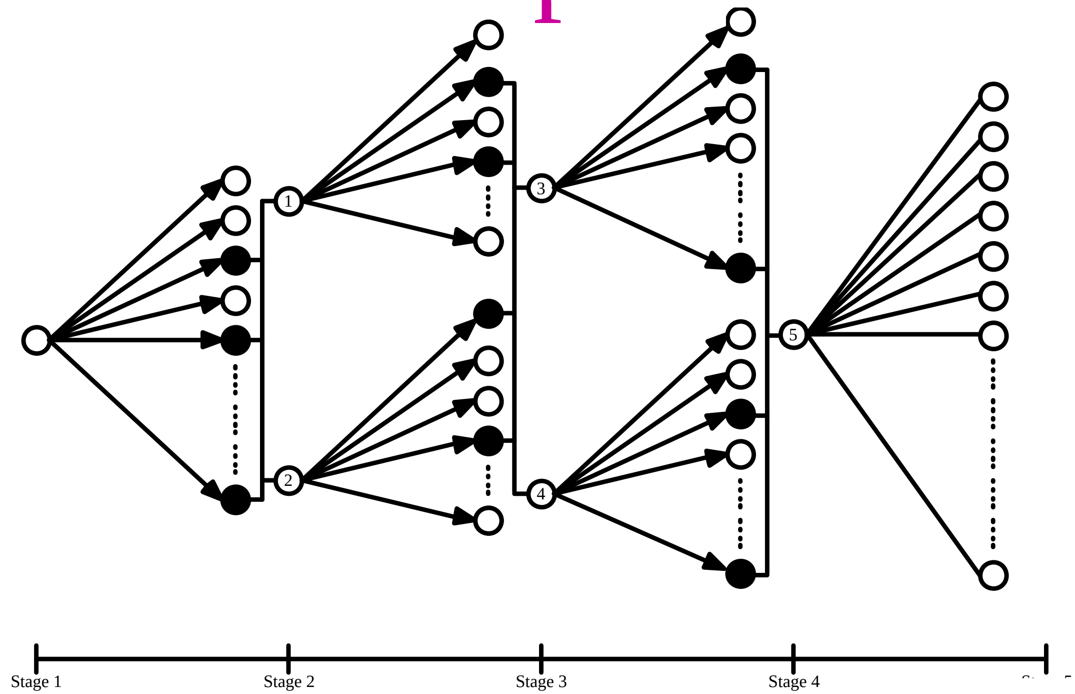
# Abridged Nested Decomposition

- Also incorporates sampling into the general framework of Nested Decomposition
- Also assumes relatively complete recourse and serial independence
- Samples both the subproblems to solve and the solutions to continue from in the forward pass

# Abridged Nested Decomposition

**Forward Pass**

1. **Solve root node subproblem**
2. **Sample Stage 2 subproblems and solve selected subset**
3. **Sample Stage 2 subproblem solutions and branch in Stage 3 only from selected subset (i.e., nodes 1 and 2)**

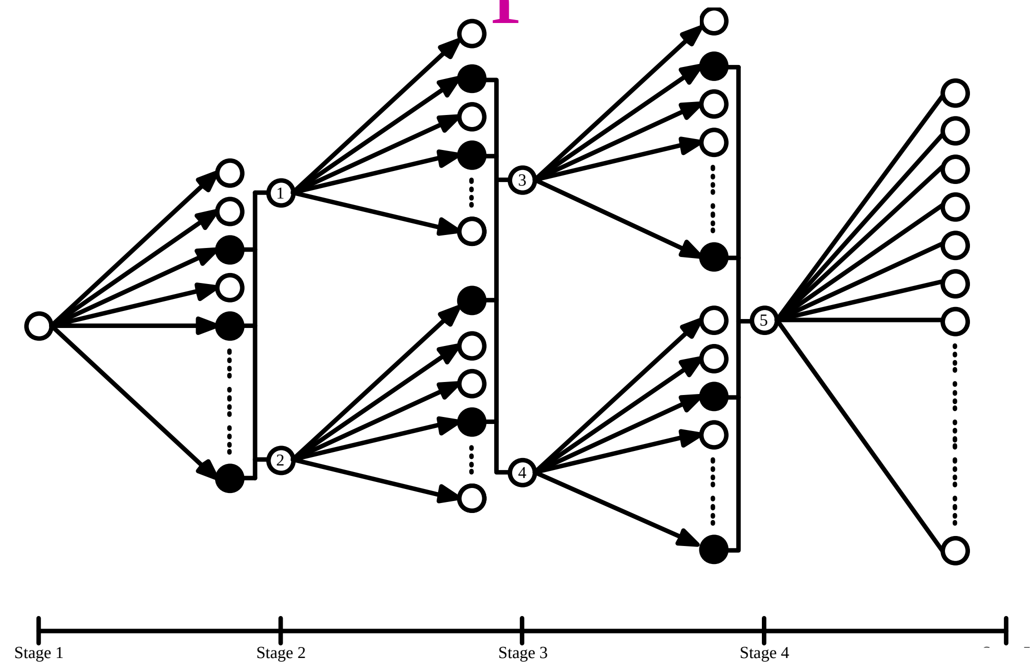



4. For each selected Stage t-1 subproblem solution, sample Stage t subproblems and solve selected subset
5. Sample Stage t subproblem solutions and branch in Stage t+1 only from selected subset

# Abridged Nested Decomposition

**Backward Pass**

1. **Starting in first branching node of Stage t = N-1, solve all Stage t+1 descendant nodes and construct new optimality cut for all stage t subproblems. Repeat for all sampled nodes in Stage t, then repeat for t = t - 1**



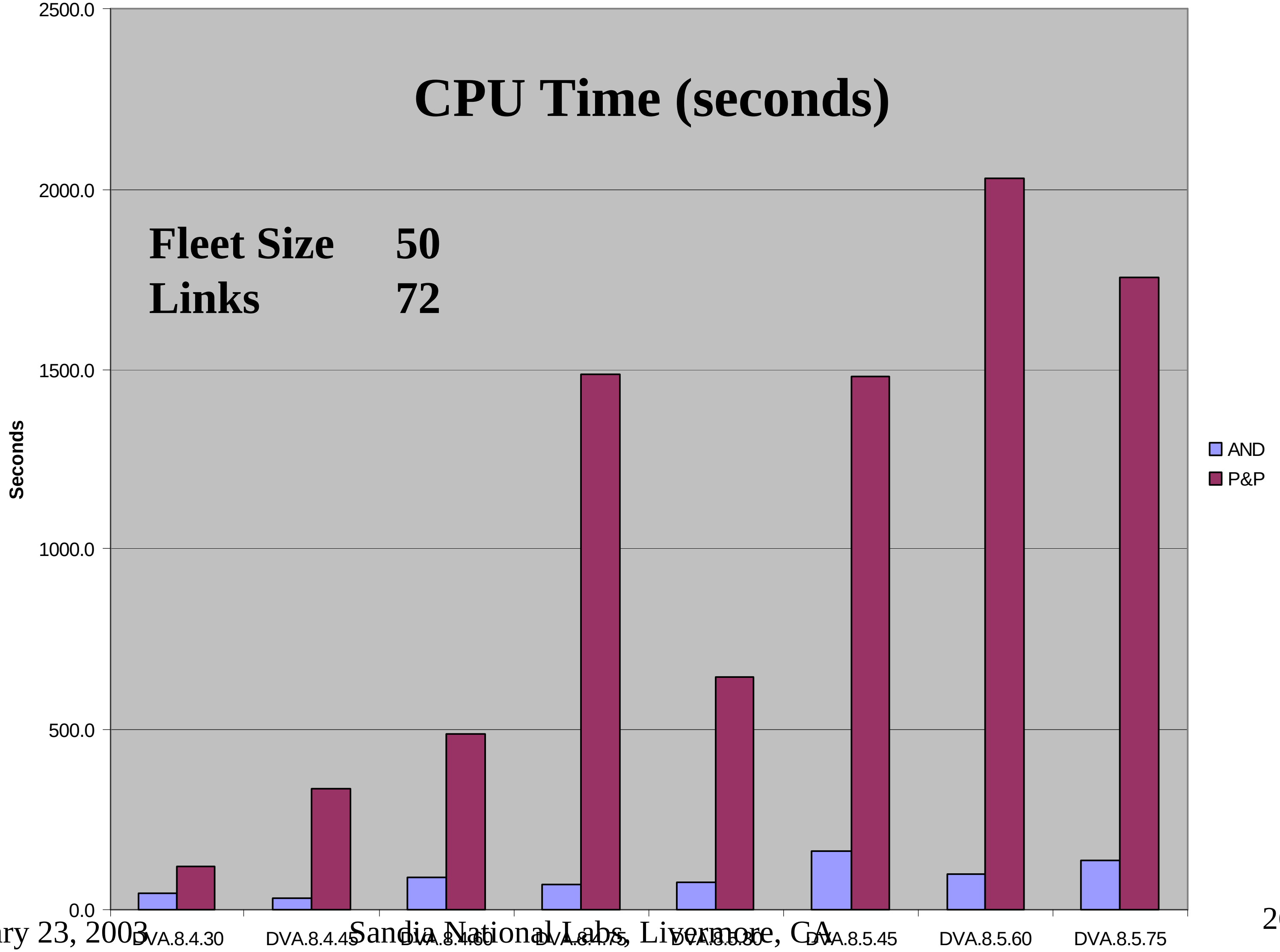

Convergence Test

1. Randomly select *H N*-Stage scenarios. For each sampled scenario, solve subproblems from root to leaf to obtain total objective value for scenario
2. Calculate statistical estimate of the first stage objective value $\bar{z}$
   – algorithm terminates if current first stage objective value $c_1x_1 + \theta_1$ is within a specified confidence interval of $\bar{z}$, else, a new forward pass begins

# Sample Computational Results

- Test Problems
  - Dynamic Vehicle Allocation (DVA) problems of various sizes
    - set of homogeneous vehicles move full loads between set of sites
    - vehicles can move empty or loaded, remain stationary
    - demand to move load between two sites is stochastic
  - DVA.*x*.*y*.*z*
    - $x$ number of sites (8, 12, 16)
    - $y$ number of stages (4, 5)
    - $z$ number of distinct realizations per stage (30, 45, 60, 75)
  - largest problem has > 30 million scenarios

# Computational Results (DVA.8)



# Outline

- General Model – Observations
- Overview of approaches
- Factorization (interior point/barrier)
- Decomposition
- Lagrangian methods
- Conclusions

# Lagrangian-based Approaches

- General idea:
  - Relax nonanticipativity
  - Place in objective
  - Separable problems

MIN $\quad E\,[\,\Sigma_{t=1}^{T} f_t(x_t,x_{t+1})\,]$
s.t. $\quad x_t \in X_t$
$\quad x_t$ nonanticipative

MIN $\quad E\,[\,\Sigma_{t=1}^{T} f_t(x_t,x_{t+1})\,]$
$x_t \in X_t$
$\quad + E[\underline{w}_t x] + r/2||x-\underline{x}||^2$

**Update**: $w_t$; **Project**: x into N - nonanticipative space as $\underline{x}$

Convergence: Convex problems - Progressive Hedging Alg.
(Rockafellar and Wets)
Advantage: Maintain problem structure (networks)

# Lagrangian Methods and Integer Variables

- Idea: Lagrangian dual provides bound for primal but
  - Duality gap
  - PHA may not converge
- Alternative: standard augmented Lagrangian
  - Convergence to dual solution
  - Lose separability
  - May obtain simplified set for branching to integer solutions
- Problem structure: Power generation problems
  - Especially efficient on parallel processors
  - Decreasing duality gap in number of generation units

# Outline

- General Model – Observations
- Overview of approaches
- Factorization (interior point/barrier)
- Decomposition
- Lagrangian methods
- Conclusions

# Some Open Issues

- Models
  - Impact on methods
  - Relation to other areas
- Approximations
  - Use with sampling methods
  - Computation Constrained Bounds
  - Solution Bounds
- Solution methods
  - Exploit specific structure
  - Parallel/distributed architectures
  - Links to approximations

# Criticisms

- Unknown costs or distributions
  - Find all available information
  - Can construct bounds over all distributions
    - Fitting the information
  - Still have known errors but alternative solutions
- Computational difficulty
  - Fit model to solution ability
  - Size of problems increasing rapidly

# Conclusions

- Stochastic programs structure:
  - repeated problems
  - nonzero pattern for sparsity
  - use of decomposition ideas
- Results
  - take advantage of the structure
  - speedups of orders of magnitude